

Application of Artificial Neural Networks for the Simulation of a Perfectly Premixed Chemical Reactor

Florian Setzwein*, Martin Grader†, Timo Seitz‡, Peter Ess§ and Peter Gerlinger¶

Institute of Combustion Technology, German Aerospace Center (DLR), Stuttgart, 70569, Germany

Finite-rate chemistry combustion simulations require the computationally expensive direct integration of the chemical source term. In order to reduce the calculation time of such simulations, an efficient artificial neural network approach is developed. The artificial neural network consists of a classification methodology to subdivide the thermochemical state space into smaller clusters and a subsequent multi-layer perceptron for each of the clusters. Due to the subdivision approach, the multi-layer perceptron size can be kept small, which guarantees a computationally efficient artificial neural network. For the classification two different approaches are investigated, namely a self-organizing map and a k -means binary decision tree. Both methods are tested with respect to their clustering quality and their performance. Chemical reactors are used to generate training and validation data. The multi-layer perceptron prediction includes all species of the original chemical mechanism. A priori comparison with direct integration proved the ability of both methods to give accurate results even for minor species. A posteriori calculations of ignition delay are conducted over an initial temperature range of 1100 to 1700 K and an equivalence ratio range of $\phi = 0.7$ to $\phi = 1.4$. While species concentrations and temperature profiles are reproduced well for most of the initial conditions, the prediction quality of the artificial neural networks decreases for a few calculations starting at low temperatures. Performance benchmarks confirmed that the artificial neural network approaches are superior to direct source term integration in terms of computational costs. The benchmarks also revealed that the k -means binary decision tree-based approach is three times faster than the self-organizing map approach.

I. Introduction

Computational Fluid Dynamics (CFD) plays an important role in the development of gas turbine and rocket engine combustors. Due to the increasingly strict pollutant regulations and efficiency requirements, more accurate numerical simulations are required. Such simulations are very expensive due to the need to resolve unsteady phenomena like thermo acoustics or to include the formation of pollutants. The major cost factor in combustion simulations is the integration of the chemical source term which describes the production or consumption of a chemical species. Especially the prediction of pollutants is very sensitive and relies on an accurate prediction of chemical species and their sources. Simple, yet efficient combustion models facilitate or avoid the integration of the source term by making assumptions, which restrict the generality of the model. More universal models, which directly integrate the chemical source term during runtime, come with a drastic increase in computational costs. To bridge the gap between costs and accuracy, Artificial Neural Networks (ANNs), which are trained to reproduce the integration of the chemical source term, promise a significant cost reduction while maintaining a good predictive accuracy.

The past decade has seen a massive grow in computational power spent on artificial intelligence applications.¹ The latter have also found their way into combustion simulations to e.g. reduce memory requirements

*PhD Student, corresponding author, florian.setzwein@dlr.de

†PhD Student

‡PhD Student

§Postdoctoral Fellow, AIAA Senior Member

¶Professor

of tabulated approaches,²⁻⁴ close unresolved turbulence chemistry interaction^{5,6} or increase computational efficiency.⁷⁻¹⁵ The possible acceleration rates depend on the implementation, the reaction mechanism size and the topology of the ANN. Application in a Direct Numerical Simulation (DNS)¹⁵ for example showed an acceleration by a factor of 25, whereas partially stirred reactor simulations have been accelerated by a factor of up to 1250.⁹ Application in a combined Large-Eddy Simulation (LES) - Transported Probability Density Function (TPDF) approach¹³ achieved an acceleration of a factor of 83. The fact that no major accuracy issues occurred highlights the potential of ANNs to make sophisticated combustion models affordable in CFD. A crucial aspect in using ANNs is their training and in particular the quality of the training data set. In order to capture combustion phenomena well, the data used to train the ANN has to cover the whole thermochemical state space which is present in the target flame. Training data is usually generated by simulations that directly integrate the chemical source term. The calculations can either be performed for the application case or for simpler, yet still representative, generic flames. Only the latter approach allows for a universal application of the ANN based simulation. Such generic training data can be extracted from DNS,^{10,15} laminar flames^{4,12-14} or chemical reactor calculations.^{4,9} In this work we focus on the latter due to its simplicity and flexibility.

There are two strategies that are encountered most frequently for the calculation of the chemical source term by means of ANNs. The first approach is based on the subdivision of the thermochemical state space. This can either be achieved via simple constraints such as high and low temperature regions, or alternatively through sophisticated neural network structures such as a Self-Organizing Map (SOM).¹⁶ The latter are designed to reduce a high-dimensional space into a one- or two-dimensional subspace and have been used, for example, in the works of Blasco et al.⁹ and Franke et al.¹³ Recently, Nguyen et al.¹⁷ used *k*-means clustering¹⁸ for the subdivision task, which is another widely used algorithm for dividing datasets into different groups. After the subdivision process, the calculation of the chemical source term is carried out for each part of the state space by a separate neural network. This generally involves the use of Multi-Layer Perceptrons (MLPs), which are feedforward neural networks that are capable to reproduce the non-linearity of the chemical source term. The combination of a clustering approach and MLPs has been shown to provide good accuracy^{9,13,17} and seems to be a promising candidate in terms of computational effort, because the size of the MLPs can be kept small. The second approach that is frequently encountered for the calculation of the chemical source term using ANNs are single network units that cover the entire thermochemical state space. This methodology requires much more complex MLP structures such as Deep Neural Networks (DNNs) or Convolutional Neural Networks (CNNs). The capability of this approach was for example demonstrated by Wan et al.¹⁵ for DNS of a turbulent syngas oxy-flame and by Zhang et al.¹⁹ for the homogeneous autoignition of dimethyl ether.

In this work, the first approach is used, where the subdivision of the thermochemical state space is carried out with a SOM and the chemistry integration is performed with subsequent MLPs. This entire structure of networks is referred to as ANN compound. Besides the application of SOMs, a novel clustering strategy is introduced, which consists of a binary decision tree that is constructed through the *k*-means clustering approach. Due to the tree structure, this method should work much more efficiently than for example SOMs. A similar data structure has recently been used by Wang et al.²⁰ for the classification of several publicly available datasets. However, to our knowledge, this method has not been used for the clustering and classification of thermochemical states in the context of combustion simulations. Both clustering methods are investigated in terms of accuracy and computational time for homogeneous reactor simulations.

II. Methods

In addition to the conservation equations of mass, energy and momentum, detailed reactive flow simulations require the solution of a coupled system of transport equations for N_{sp} involved chemical species α . Considering Einstein's summation convention, these transport equations can be written as

$$\frac{\partial}{\partial t} (\rho Y_\alpha) + \frac{\partial}{\partial x_i} (\rho u_i Y_\alpha) + \frac{\partial j_{i,\alpha}}{\partial x_i} = S_\alpha(\Phi), \quad \alpha = 1, \dots, N_{sp}. \quad (1)$$

The left side of Eq. (1) describes the convective and diffusive transport of the species mass fraction Y_α of a species α in a fluid with density ρ and the velocity vector u_i . The term $j_{i,\alpha}$ denotes the diffusion mass flux of α in the x_i -direction. The chemical source term $S_\alpha(\Phi)$ reflects the production and consumption of a species α by chemical reactions. It is calculated from a reaction mechanism, which typically comprises a

large number of elementary reactions. The source term varies in a highly nonlinear manner with the species mass fractions Y_α and temperature T , which are generally summarized in the thermochemical state vector $\Phi = [p, T, Y_1, Y_2, \dots, Y_{N_{sp}}]$. However, p is excluded from the state vector Φ in the following analysis, since an isobaric system is considered. The last species $Y_{N_{sp}}$ is excluded as well, since it can be deduced from the conservation of mass $\sum_\alpha Y_\alpha = 1$. A detailed description of the numerical solution of such combustion problems can be found in the literature.^{21,22} Accurate chemical kinetics simulations of most fuels require the consideration of many species. Consequently, a large number of transport equations must be solved, which requires a considerable computational effort, especially for the calculation of the source term $S_\alpha(\Phi)$. Since neural networks are based on nonlinear activation functions, they offer the possibility to approximate the highly nonlinear processes in combustion problems with good accuracy. Besides this, the evaluation through neural networks requires only little computational effort, since it mostly involves matrix multiplications. The following sections describe a methodology for the approximation of state changes of a combustion system with an ANN compound in order to obtain an indirect tabulation of thermochemical states.

A. Preliminaries

In the following, the spatial transport and the diffusion of chemical species in Eq. (1) are neglected. This results in a perfectly premixed reactor model, where the production and consumption of a species α is described by an initial value problem for a system of ordinary differential equations

$$\frac{d\mathbf{Y}}{dt} = \frac{1}{\rho} \mathbf{S}(\Phi). \quad (2)$$

Starting from an initial state $\Phi(t)$ the mass fractions at a future time step $\mathbf{Y}(t + \Delta t)$ can be calculated by the numerical integration of the system of differential equations

$$\Delta \mathbf{Y} = \int_t^{t+\Delta t} \frac{1}{\rho} \mathbf{S}(\Phi) d\tau, \quad (3)$$

where $\Delta \mathbf{Y} = \mathbf{Y}(t + \Delta t) - \mathbf{Y}(t)$ refers to the change in species mass fraction over a time step Δt . The present approach for a neural network solver is to determine an approximation function $F_{\text{MLP}}(\Phi; \Delta t)$ in the form of a MLP model, which is capable of approximating the integral for a defined time step Δt :

$$F_{\text{MLP}}(\Phi; \Delta t) \approx \int_t^{t+\Delta t} \frac{1}{\rho} \mathbf{S}(\Phi) d\tau. \quad (4)$$

A MLP is a feedforward neural network that is composed of several layers of nodes. It acts as a multidimensional regression function, which consists of nested applications of nonlinear activation functions. Hence, it can be used to reproduce nonlinear trajectories in the chemical state space, which are encountered in combustion. A more detailed explanation of the utilized MLP structure is given below in the text. Approximating the chemical source term integral (4) with a well-trained neural network would require very little computational effort, since it mostly involves matrix multiplications. Thus, it offers a significant speedup for the estimation of a composition $\mathbf{Y}(t + \Delta t)$ compared to the numerical integration of Eq. (3). The unknown weights of the function $F_{\text{MLP}}(\Phi; \Delta t)$ have to be estimated as the solution of an optimization problem using predetermined thermochemical states. This is referred to as a supervised training process of the neural network. However, training of a single MLP over a wide range spanned by the thermochemical state space turns out to be difficult, since the changes of different species usually have to be approximated with sufficient accuracy over several orders of magnitude. As a result, the optimization of the function weights converges very slowly and thus complicates the overall training process. A more promising approach is the division of the available training states into different clusters and the subsequent training of a composite of several MLPs $F_{\text{MLP},i}(\Phi; \Delta t)$. Each of these networks belongs to one cluster and approximates only a sub-region of the thermochemical state space. This improves the convergence of the training process and the accuracy of the approximation significantly. Finally, the methodology for regression of the chemical source term using neural networks can be divided into three parts:

1. Generation of training data using direct numerical integration for a specified numerical time step Δt .
2. Subdivision of the obtained training states into different clusters based on their position in the thermochemical state space.

3. Training of a single MLP $F_{\text{MLP},i}(\Phi; \Delta t)$ for each cluster.

Once all networks are trained and validated, the temporal integration of the system of equations can be approximated with the entire compound of networks. In order to do so, first a suitable network $F_{\text{MLP},i}(\Phi; \Delta t)$ must be selected for a corresponding input state $\Phi(t)$, before the new species composition $\mathbf{Y}(t + \Delta t)$ is calculated. In contrast to classical approaches, where the entire state space is tabulated, this methodology has very low memory requirements, since only the MLP weights and the composite structure of the networks must be stored. This also results in a methodology which allows easy parallelization.

B. Training data generation

Suitable data for the network training is generated by the numerical integration of the system of equations given in Eq. (3). For this purpose, an isobaric, adiabatic, perfectly premixed reactor model is employed. The resulting system of equations to be solved has the following form:

$$\frac{\partial Y_\alpha}{\partial t} = \frac{W_\alpha}{\rho} \dot{\omega}_\alpha(\Phi), \quad \alpha = 1, \dots, N_{sp}. \quad (5)$$

Here, W_α represents the molecular weight and $\dot{\omega}_\alpha(\Phi)$ the production rate of species α . The latter is a nonlinear function, which depends on the temperature T , the species mass fractions Y_α , and the chemical kinetics of the underlying reaction mechanism. An adiabatic and isobaric system is assumed, which allows the temperature to be derived from the caloric equation of state on the condition that $\partial h / \partial t = 0$.

The entire system of equations is advanced in time for different initial equivalence ratios ϕ_0 of fuel and oxidizer and for different starting temperatures T_0 . Sufficiently small integration time steps Δt are required to ensure numerical stability. Thus, a few variations of initial conditions already result in a large number of data samples $N_{s,\text{total}}$, if every single state along a trajectory is chosen. To prevent an over-representation of certain regions in the state space and to reduce the overall amount of data, only a subset of all generated samples is used. For this, the temperature and all species along all generated trajectories are normalized to the range between zero and one. This normalization consists of three steps, which are carried out in the following way:

$$\bar{\Phi}_i(t_s) = \frac{\Phi_i(t_s) - \Phi_i^{(\min)}}{\Phi_i^{(\max)} - \Phi_i^{(\min)}}, \quad \tilde{\Phi}_i(t_s) = \log_{10} \left(\frac{1 + \bar{\Phi}_i^\kappa(t_s)}{\kappa} \right), \quad \hat{\Phi}_i(t_s) = \frac{\tilde{\Phi}_i(t_s) - \tilde{\Phi}_i^{(\min)}}{\tilde{\Phi}_i^{(\max)} - \tilde{\Phi}_i^{(\min)}}, \quad (6)$$

where $\Phi(t_s)$ refers to a single state space sample at time t_s . The values of $\Phi^{(\min)}$ and $\Phi^{(\max)}$ denote the minimum and maximum values of all generated states, respectively. A factor of $\kappa = 0.1$ is chosen for the logarithmization, which has also been utilized in the work of Zhang et al.¹⁹ Note, that the logarithmization step is not applied for the temperature. Next, only those samples are selected for the upcoming training, which differ sufficiently from their preceding state. The selection is based on the vector norm of two consecutive normalized states with respect to a preset threshold $\varepsilon = 10^{-5}$. This results in the final set of training samples S , which is defined by

$$S = \left\{ \Phi^{(s)} : \left\| \hat{\Phi}(t_s + \Delta t) - \hat{\Phi}(t_s) \right\|_2 > \varepsilon, \quad s = 1, \dots, N_{s,\text{total}} \right\}. \quad (7)$$

The resulting training data for a representative trajectory is shown as an example in Fig. 1.

C. Training data subdivision

Once the training data has been generated and reduced, the samples must be clustered before they can be used as input data for the training of the MLPs $F_{\text{MLP},i}(\Phi, \Delta t)$. One possible approach for this task would be a simple cartesian subdivision of the thermochemical state space. However, the selected training states from the perfectly premixed reactor merely represent a subset of the entire thermochemical state space. Moreover, they also cover only a small fraction of all possible states that would occur, for example, in technical flames. Thus, a cartesian subdivision would result in a high number of regions that are not covered by any training samples. Likewise, a few regions would contain many samples, which would suppress the advantages of the subdivision. It is much more effective to group only the available dataset of training states into several clusters. This can be addressed using an unsupervised learning process, where no target

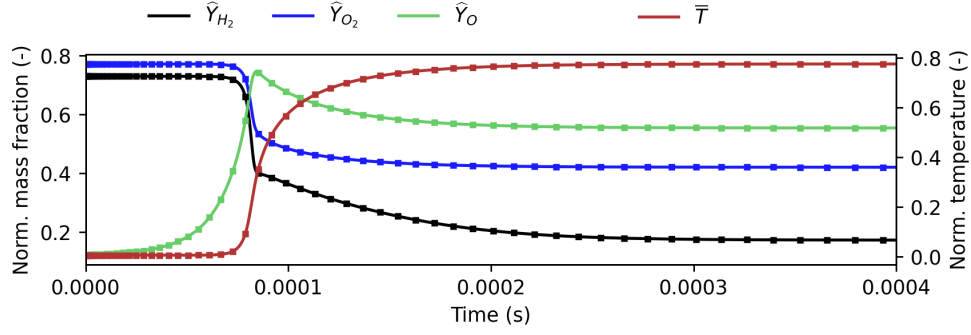


Figure 1: Exemplary trajectory of a perfectly premixed reactor calculation. The points represent extracted training samples S used for training the networks. Only every tenth sample is shown for reasons of clarity.

variables are provided for clustering the high-dimensional training data. In this work, this task is realized with two different approaches, namely SOMs and k -Means Binary Decision Trees (KMBDTs). Both methods are described in detail in the following sections.

1. Self-Organizing Map

A SOM is a neural network type that consists of a single input and a single output layer. It was introduced by Kohonen^{16,23} and can be used to transform high-dimensional data into a structured representation. In this work, the network input nodes refer to the features of the normalized N_{sp} -dimensional input state vector $\hat{\Phi} = [\bar{T}, \hat{Y}_1, \dots, \hat{Y}_{N_{sp}-1}]^T$. The nodes in the output layer are arranged in a two-dimensional lattice and each node i is associated to a vector of weights $\mathbf{w}_i = [w_{i,1}, w_{i,2}, \dots, w_{i,N_{sp}}]^T$. The training process involves a consecutive presentation of training samples to the SOM. In every training iteration s , all output nodes compete for a presented training input sample $\hat{\Phi}^{(s)}$. The node $n^{(s)}$, that features the most similar weights $\mathbf{w}_n^{(s)}$ compared to $\hat{\Phi}^{(s)}$, is referred to as the best-matching unit.²⁴ Its index is estimated through

$$n^{(s)} = \arg \min_i \left\| \hat{\Phi}^{(s)} - \mathbf{w}_i^{(s)} \right\|_2, \quad i = 1, 2, \dots, N_{cl}, \quad (8)$$

where N_{cl} refers to the number of clusters in the SOM output nodes in total. The weights of the best-matching unit and its immediate neighbors are then subsequently adapted through

$$\mathbf{w}_i^{(s+1)} = \mathbf{w}_i^{(s)} + \eta^{(s)} h_{i,n}^{(s)} \left(\hat{\Phi}^{(s)} - \mathbf{w}_i^{(s)} \right), \quad (9)$$

with a learning rate $\eta^{(s)}$ and a neighborhood function $h_{i,n}^{(s)}$, both of which change along with the number of iterations s . The latter can be defined through²⁴

$$h_{i,n}^{(s)} = \exp \left(-\frac{\|\mathbf{r}_i - \mathbf{r}_n\|_2}{2\pi (\sigma^{(s)})^2} \right), \quad (10)$$

where \mathbf{r}_i and \mathbf{r}_n refer to the lattice position of output node i and $n^{(s)}$ in the two-dimensional SOM structure, respectively. The effective width $\sigma^{(s)}$ of the neighborhood gets adjusted in every iteration according to²⁴

$$\sigma^{(s+1)} = \frac{s_{\max}}{2s + s_{\max}} \sigma^{(s)} \quad (11)$$

and the learning rate $\eta^{(s)}$ is adjusted in a similar way. A more detailed description of SOM can be found in the literature.^{23–26} The training process is finished when a maximum iteration s_{\max} is reached. Figure 2 shows an example SOM network on the left, which is trained for a generic two-dimensional input dataset, as well as the interconnectivity between the output nodes. A trained SOM network is able to assign a

corresponding best-matching unit to an input vector which is not contained in the training dataset. This is referred to as classification. The usage of a SOM is twofold within this work. First, the network is used to partition the reduced training dataset S into N_{cl} subsets of state samples, which ideally should feature similar sizes. Every SOM output node i is then connected to a separate MLP $F_{MLP,i}(\Phi, \Delta t)$, which is trained with the respective subset samples. Once the training is completed, the trained SOM network is used in the final ANN compound to select an appropriate MLP $F_{MLP,i}(\Phi, \Delta t)$ for a given input state Φ .

2. k -Means Binary Decision Tree

The determination of the two-dimensional SOM structure can be very time-consuming for large datasets. It is also possible that the number of distributed samples may vary significantly for different clusters. Besides this, to find a best-matching unit in a trained SOM network, it is required to traverse every single SOM output node. This can become time-consuming if a large SOM structure is employed. We propose a different data structure for the clustering which intends to overcome these shortcomings and which is referred to as KMBDT. In this approach, a cluster of training data is divided recursively into two further sub-clusters, as long as these do not fall below a prescribed number of samples. Every sub-cluster is connected to its parent, in order to create a binary decision tree structure in a top-down manner. An example of a corresponding division tree structure is given in Fig. 2 on the right. The splitting operation is based on the k -means clustering approach, which is applied for every sub-cluster. The k -means clustering¹⁸ is a well-established strategy for the unsupervised partitioning of a set of higher-dimensional sample vectors into k clusters. In the current approach, the partitioning is performed for $k = 2$. Similarly to SOM, every cluster i is associated to a weight vector \mathbf{w}_i . These weights can be chosen randomly at the beginning of the training process. During a training iteration s , all training samples $\hat{\Phi}$ in S are distributed among k subsets $S_i^{(s)}$, each of which is associated to a cluster i . The distribution is based on the current cluster weights \mathbf{w}_i such that

$$S_i^{(s)} = \left\{ \hat{\Phi}^{(p)} : \left\| \hat{\Phi}^{(p)} - \mathbf{w}_i^{(s)} \right\|_2^2 < \left\| \hat{\Phi}^{(p)} - \mathbf{w}_j^{(s)} \right\|_2^2, \quad j = 1, \dots, k, \quad i \neq j \right\}. \quad (12)$$

Once the distribution is completed, the cluster weights are updated according to

$$\mathbf{w}_i^{(s+1)} = \frac{1}{N_{s,i}} \sum_{\hat{\Phi}^{(p)} \in S_i^{(s)}} \hat{\Phi}^{(p)}, \quad (13)$$

where $N_{s,i}$ refers to the number of samples in $S_i^{(s)}$. This process is repeated until the weights do not change any more. Again, for a detailed description on the k -means algorithm, we refer to the literature.^{18, 25–28} Upon the classification stage, the tree is traversed along the nodes, whose weights feature the smallest Euclidean distance to a given input sample. On average, the lookup time scales with $\mathcal{O}(\log_2(N_{cl}))$, where N_{cl} refers to the total number of clusters in the tree. For comparison, the lookup time for the SOM approach scales with $\mathcal{O}(N_{cl})$, and is thus considerably higher for large networks. Similar to the SOM approach, the KMBDT structure is used for both clustering the training data and classifying input states for the trained ANN compound.

D. Neural network training

Once the data is clustered, either using a SOM or a KMBDT, the training of the actual nonlinear MLPs is conducted. In the SOM structure, every output node and its associated training data is assigned to a network $F_{MLP,i}(\Phi; \Delta t)$, whereas in the KMBDT, this is done only for the leaf nodes. A MLP is basically a nonlinear multidimensional regression model that maps an input vector \mathbf{x} onto an output vector \mathbf{y} . It consists of an input and an output layer which are connected by one or more intermediate layers. The latter are also referred to as hidden layers.²⁵ Each of these layers consists of a number of nodes, which may differ from layer to layer. A schematic MLP is shown in Fig. 3. At each node of the intermediate and output layers, the following operation is performed:

$$y_{p+1,q} = g \left(\sum_{r=1}^{N_{lp}} w_{p,q,r} x_{p,r} + w_{p,q,0} \right). \quad (14)$$

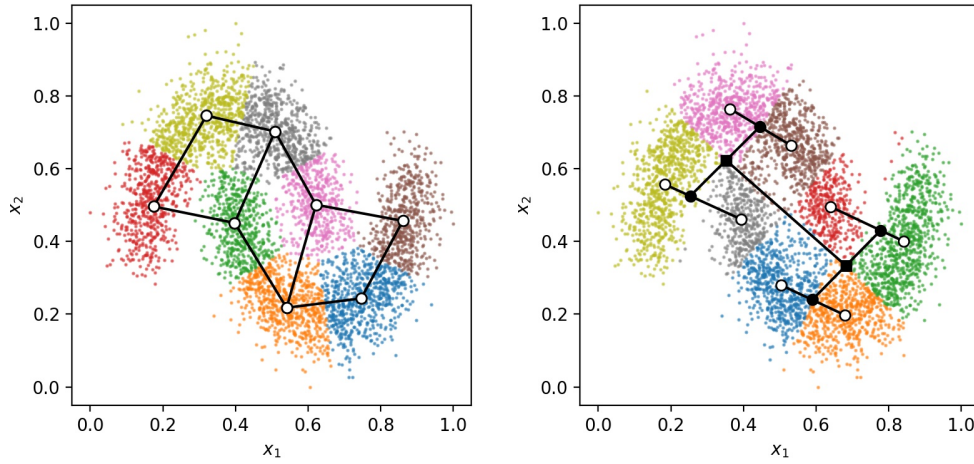


Figure 2: Example clustering for a generic two-dimensional dataset using the SOM approach (left) and the KMBDT (right). The black segments highlight the interconnectivity between network nodes. For the KMBDT, white dots refer to leaf nodes and black dots denote internal nodes.

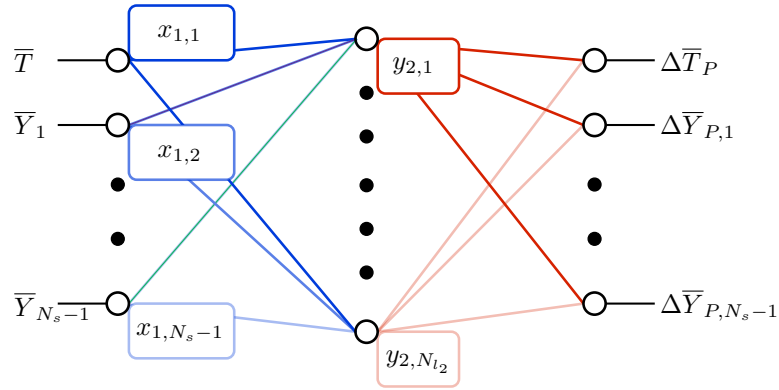


Figure 3: Schematic structure of a MLP with a single intermediate layer. The connections between nodes represent the unknown coefficients $w_{p,q,r}$.

Here, $y_{p+1,q}$ is the output value of node q in layer p , which comprises a total of N_{lp} nodes. The input value $x_{p,r}$ originates from a node r in a previous layer, which is connected to node p . This input data is weighted by the unknown coefficients $w_{p,q,r}$ and an offset factor $w_{p,q0}$. The result is then passed to the nonlinear activation function $g(\mathbf{x})$. The number of intermediate layers and the number of nodes in each intermediate layer represent the major degrees of freedom in generating a MLP. Correspondingly, networks with a higher number of nodes and layers are able to capture more complex relationships in the datasets. In this work, the state vectors $\bar{\Phi}^{(s)}$ are used as network inputs, which are normalized with their cluster minima and maxima. The network aims to predict a corresponding normalized difference between two states $\Delta\bar{\Phi}^{(s)} = \bar{\Phi}(t_s + \Delta t) - \bar{\Phi}(t_s)$. During the training process, defined input values are presented to the network and the corresponding network output $\Delta\bar{\Phi}_{\text{ANN}}^{(s)}$ is compared with known target values $\Delta\bar{\Phi}_{\text{Target}}^{(s)}$. The latter stem from the numerical integration of the system of equations (3). The coefficients $w_{p,q,r}$ and $w_{p,q0}$ are adjusted iteratively using a gradient descent optimization, such that the following loss function is minimized for all $N_{s,i}$ samples in the training dataset:²⁹

$$E = \frac{1}{2} \sum_{s=1}^{N_{s,i}} \left\| \Delta\bar{\Phi}_{\text{ANN}}^{(s)} - \Delta\bar{\Phi}_{\text{Target}}^{(s)} \right\|_2^2 + P\left(\bar{\Phi}_{\text{ANN}}^{(s)}\right). \quad (15)$$

The penalty term $P\left(\Phi_P^{(s)}\right)$ is defined as

$$P\left(\Phi_P^{(s)}\right) = -\min\left(0, T_P^{(s)}\right) - \sum_{\alpha=1}^{N_{sp}-1} \min\left[0, \left(1 + \Delta Y_{P,\alpha}^{(s)} - Y_{\alpha}^{(s)}\right)\right] - \sum_{\alpha=1}^{N_{sp}-1} \min\left[0, \left(Y_{\alpha}^{(s)} - \Delta Y_{P,\alpha}^{(s)}\right)\right] \quad (16)$$

It aims to prevent the network from being trained for non-physical mass fractions or negative temperatures. Two different activations functions are employed in this work. The softsign function is used for hidden network layers

$$g_{\text{softsign}}(x) = \frac{x}{1 + |x|}, \quad (17)$$

whereas values from the output layer are weighted and passed through a sigmoid function

$$g_{\text{sigmoid}}(x) = \frac{1}{1 + \exp(-x)}. \quad (18)$$

For a more comprehensive description on the MLP training we refer to the literature.^{25, 26, 29} Finally, the fully trained ANN compound can be used to approximate Eq. (3) through the following steps:

1. Normalize a given state vector $\Phi^{(s)}$ to get $\hat{\Phi}^{(s)}$.
2. Use $\hat{\Phi}^{(s)}$ to select the best-matching MLP by using either the SOM or the KMBDT approach.
3. Normalize the state vector $\Phi^{(s)}$ again to obtain $\bar{\Phi}^{(s)}$.
4. Use $\bar{\Phi}^{(s)}$ to calculate $\Delta\bar{\Phi}^{(s)}$ with the best-matching MLP.
5. Denormalize $\Delta\bar{\Phi}^{(s)}$ to get $\Delta\Phi^{(s)}$.
6. Calculate the state vector at the next time step as $\Phi(t_s + \Delta t) = \Phi^{(s)} + \Delta\Phi^{(s)}$.

Note that the different normalization procedures are necessary to achieve a good separation of the state space on the one hand and a good accuracy of the MLP calculations on the other hand.

III. Application of ANN methodology

In this section, the methodology introduced above will be applied to well stirred chemical reactors of hydrogen and oxygen. First, the mechanism, the test dataset and the results of the two data clustering approaches are introduced. Next, an a priori comparison of the SOM-based and the KMBDT-based ANN compounds is performed, followed by an a posteriori validation as a basis to check their ability to perform as a chemical reactor. Finally, the computational costs of both ANN approaches are compared to the costs of direct integration.

A. Setup

In order to demonstrate the applicability of the presented method, a reduced hydrogen oxidation mechanism^{30, 31} is utilized. This mechanism consists of 21 reversible reactions as well as the eight reactive species $\text{H}_2, \text{O}_2, \text{H}_2\text{O}, \text{H}, \text{O}, \text{OH}, \text{HO}_2$ and H_2O_2 . The training data is generated from adiabatic, isobaric, perfectly premixed reactor calculations using the Cantera software³² for direct integration. A total of 3600 initial conditions of the reactors are distributed over a temperature range between $T = 900 \text{ K}$ to $T = 1700 \text{ K}$ as well as an equivalence ratio interval between $\phi = 0.5$ to $\phi = 1.7$. As indicated in Fig. 4, a large fraction of the starting conditions is located in the range of low temperatures and low equivalence ratios. The perfectly premixed reactor model is integrated with a time step size of $\Delta t = 1 \times 10^{-8} \text{ s}$ over a time span of $t = 4 \times 10^{-4} \text{ s}$. This duration ensures that all simulations reach a steady state. Due to the aforementioned data reduction, only 0.7% of all created data points are used for training and testing of the ANN. In order to investigate different clustering approaches, the SOM and the KMBDT are trained with this dataset. The SOM nodes are located on a two-dimensional grid with 45×20 nodes. Some nodes are assigned to an insufficient number of samples, so that a total of 875 MLPs results. Python's MiniSom implementation³³ is utilized for SOM training. The

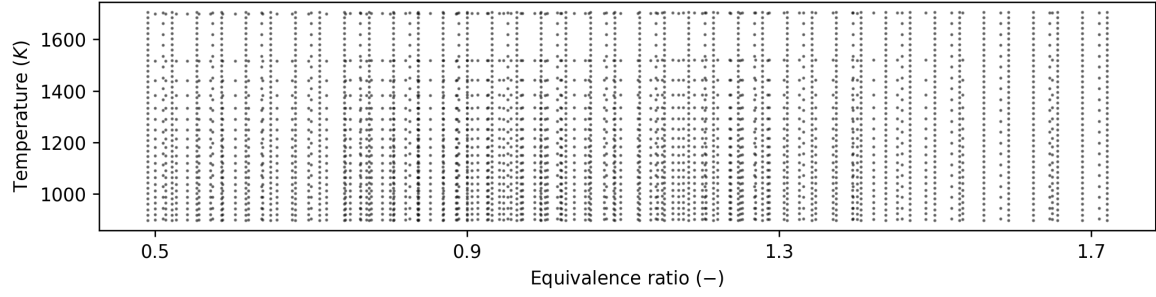


Figure 4: Starting conditions used for the training data generation, every dot represents a chosen combination of T_0 and ϕ_0 for the generation of a state space trajectory.

trained KMBDT consists of 13 layers and 1168 leaf nodes, which corresponds to the number of MLPs for the resulting ANN compound. The software package scikit learn²⁸ is used for the k -means clustering. For both methods, a minimum number of 500 samples is specified for a cluster to be valid. SOM nodes that feature less samples are joined with their nearest neighbor nodes. In the KMBDT method, a node is not split further if one of its children contains less than the prescribed 500 samples. Python's deep learning library pyTorch²⁹ is utilized for the actual MLP training process. Each MLP unit consists of two hidden layers with 15 neurons each and nine input and output layers for each species and the temperature. As activation function, the softsign function (Eq. (17)) is used for the hidden layers and the sigmoid function (Eq. (18)) for the output layer. The associated training samples for an MLP are further split into 12 batches, in order to accelerate the training convergence. The training for a single MLP is carried out for 10000 epochs, such that the loss function (Eq. (15)) features values in the range of $\mathcal{O}(10^{-5})$. The entire network compound as well as the classification procedures are implemented in the C programming language, in order to compare the present methodology to the direct integration in terms of performance.

Figure 5 shows how both clustering approaches cover the temperature-OH space of the training dataset. Both SOM and KMBDT subdivide the temperature-OH space unevenly; many MLP nodes are found either at high temperatures or at low OH mass fractions, while fewer nodes are found in between. This is due to the fact that the nine-dimensional state space is projected into two dimensions in this plot. Differences of both approaches can be seen at high temperatures and OH mass fractions. Here, the KMBDT covers the temperature-OH space more regularly, while the SOM approach assigns larger parts of the training dataset to the corresponding MLP nodes. Furthermore, the SOM approach shows outliers, to which no data can be assigned. They are caused by the necessity to define a fixed number of SOM nodes prior to the SOM training, which does not automatically match the optimal number of nodes. We decided to use slightly more nodes than necessary. This ensures that the clustering is optimal for the nodes to which data is assigned but slightly increases the calculation time. In contrast, the number of KMBDT nodes is flexibly assigned during the KMBDT training, which is why no outliers are present with this approach.

In order to quantify the quality of data clustering, two quantization errors are computed for the SOM and the KMBDT method. The mean squared error (MSE)

$$\text{MSE} = \frac{\sum_{i=1}^{N_s} \|\mathbf{x}_i - \mathbf{w}_{c(i)}\|_2}{N_s} \quad (19)$$

is used to quantify how well data is distributed within the clusters and the Davies-Bouldin index (DB)³⁴

$$\text{DB} = \frac{1}{N_{nn}} \sum_{i=1}^{N_{nn}} \max_{i \neq j} \left\{ \frac{\mu_i + \mu_j}{\|\mathbf{w}_i - \mathbf{w}_j\|_2}, j = 1, \dots, N_{nn} \right\} \quad (20)$$

quantifies how well data is distributed between the clusters. The MSE uses the distance between a data point \mathbf{x}_i and the closest node $\mathbf{w}_{c(i)}$ as a measure. N_s is the number of data points in the dataset. The closer the data points are located to the corresponding node, the better the quality of the data clustering. The DB measures how well the clustered or quantized datasets are separated from each other by means of the MSE of data points belonging to the i^{th} node μ_i and the distance of the node centers. Table 1 presents the

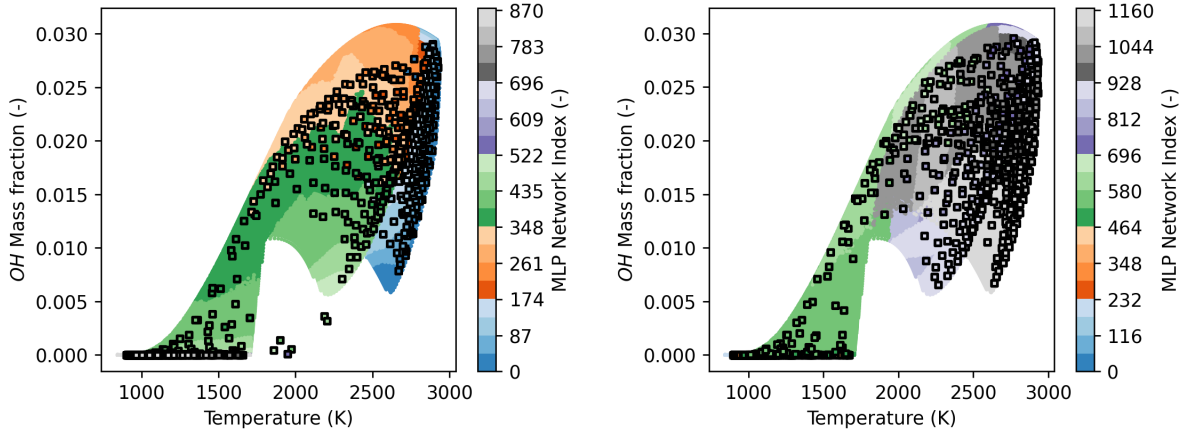


Figure 5: Representation of the nodes with associated training samples in temperature- OH space of the SOM (left) and the KMBDT (right). Each point denotes one MLP and pseudo colors indicate the node number of the MLPs.

Table 1: Quantization errors of the SOM and KMBDT approaches.

| | SOM | KMBDT |
|-----|----------|----------|
| MSE | 0.028047 | 0.022923 |
| DB | 1.4592 | 1.2139 |

accuracy of the clustering methods. The magnitude of the MSE and the DB appear reasonable for both the SOM and the KMBDT approach. This proves the success of the clustering and justifies the procedure. Furthermore, both clustering methods exhibit a similar quantization error. Hence, no method appears to be superior in terms of clustering accuracy for the chosen dataset.

B. A priori validation

The ANN compounds are validated against a test data set generated by the ODE solver. Its starting conditions cover a temperature range between $T = 1200$ K to $T = 1700$ K and an equivalence ratio interval between $\phi = 0.7$ to $\phi = 1.5$. This validation data set has not been used for the training process. In principle it is possible to use the whole $T - \phi$ range which was used to generate the training data also for validation. However, it is found that both ANN approaches tend to fail at the outer parts of the training $T - \phi$ range, especially towards low temperatures. This is mainly due to errors of the minor species H_2O_2 by a few MLP in these marginal areas. If only a small amount of H_2O_2 is present at low temperatures, its change is overestimated, ultimately leading to the failure of the ANN compounds. A quantitative comparison of both clustering methods that includes these parts would be meaningless, which is why we decided to use a narrower part of the $T - \phi$ range.

Every single data point of the validation dataset consists of gas composition and temperature and can be used as an initial condition for a single iteration of the ANN. This enables a comparison of the prediction of a single time step between the ANN and direct integration (Eq. (3)). Two criteria that assess the quality of the ANN prediction are the root mean square error normalized by the standard deviation (NRMSE) and

Table 2: NRMSE (Eq. (21)) and NMB (Eq. (22)) values for temperature and all species mass fractions.

| | NRMSE SOM | NRMSE KMBDT | NMB SOM (%) | NMB KMBDT (%) |
|-------------------------------|------------------------|-------------------------|-------------|---------------|
| T | 2.073×10^{-3} | 1.8639×10^{-3} | 0.03 | 0.03 |
| H ₂ | 5.466×10^{-3} | 5.0396×10^{-3} | 0.01 | 0.01 |
| H | 7.297×10^{-3} | 6.5217×10^{-3} | -0.01 | -0.08 |
| O ₂ | 5.414×10^{-3} | 5.3228×10^{-3} | 0.00 | -0.00 |
| OH | 5.420×10^{-3} | 5.2533×10^{-3} | -0.02 | -0.04 |
| O | 8.360×10^{-3} | 8.3038×10^{-3} | -0.01 | -0.30 |
| H ₂ O | 4.784×10^{-3} | 4.6992×10^{-3} | 0.00 | 0.00 |
| HO ₂ | 5.655×10^{-3} | 5.2805×10^{-3} | -0.04 | -0.30 |
| H ₂ O ₂ | 8.039×10^{-3} | 9.2164×10^{-3} | -0.42 | -1.34 |

the normalized mean bias (NMB). These metrics are calculated as

$$\text{NRMSE}_\alpha = \frac{\sqrt{\frac{1}{N_s} \sum_{n=1}^{N_s} \left(\Delta\Phi_{\alpha,\text{ANN}}^{(n)} - \Delta\Phi_{\alpha,\text{Target}}^{(n)} \right)^2}}{\sigma(\Delta\Phi_{\alpha,\text{Target}})} \quad \text{and} \quad (21)$$

$$\text{NMB}_\alpha = \frac{\sum_{n=1}^{N_s} \left(\Delta\Phi_{\alpha,\text{ANN}}^{(n)} - \Delta\Phi_{\alpha,\text{Target}}^{(n)} \right)}{\sum_{n=1}^{N_s} \Delta\Phi_{\alpha,\text{Target}}^{(n)}}, \quad (22)$$

where $\Delta\Phi_{\alpha,\text{ANN}}^{(n)}$ is a predicted difference between two states that stems from the ANN compound and with $\Delta\Phi_{\alpha,\text{Target}}^{(n)}$ being the corresponding target value from the direct integration. $\sigma(\cdot)$ indicates the standard deviation. Table 2 displays the NRMSE and the NMB of the SOM-ANN and the KMBDT-ANN compounds for each species. For both compounds, the temperature and the major species, i.e. hydrogen, oxygen and water, exhibit the lowest NRMSE and NMB values, whereas these values are higher for the minor species like the O- and H-radical. The highest deviation between the ANN compounds and the validation data are found for H₂O₂, which is particularly evident from the respective *NMB* values. This species differs insofar as it occurs at the lowest concentrations, which could be related to the higher errors. This points to the importance of a proper data normalization to handle the different orders of magnitudes at which species mass fractions can occur during combustion. Interestingly, both ANN compounds have very similar trends in their prediction quality, which is consistent with the previous analysis of the cluster quality. Overall, the resulting errors are low, thus indicating a very good performance of the employed ANNs.

Figure 6 compares the one-time-step prediction of the SOM-ANN with the validation data. Therefore, each validation data point is been used as a starting point for one iteration of the ANN solver. The advance in one single iteration of predicted magnitudes $\Delta\Phi_{\alpha,\text{ANN}}$ is compared to the $\Delta\Phi_{\alpha,\text{Target}}$ of all points in the validation dataset. A perfect ANN prediction would result in all predicted points laying on the reference line. As expected from the global error analysis, the temperature shows the least outliers and an almost perfect agreement with the validation data. The agreement of the combustion product H₂O is still very good, but outliers are found. The same holds for H₂O₂. All other species, which are not shown in the plots, exhibit a similar behaviour and deviate only slightly from the ideal line. Figure 7 shows the same comparison for the KMBDT-ANN. Even though the plots look slightly different, the same conclusions as for the SOM-ANN still hold. As for the global error analysis, the plots reveal the similar behaviour of the approaches. It can be concluded that, even though some outliers exist, the prediction of the both ANN compounds is accurate in this a priori validation.

C. A posteriori validation

The ability of the ANN compounds to predict the progress of a chemical reactor is evaluated in a second benchmark, which is referred to as a posteriori validation. Here, the ANN compound is only fed with a single state as an initial condition. It then must predict the temperature and species mass fractions for all subsequent time steps up to a total simulation time of $t = 4 \times 10^{-4}$ s. This test case is more complex

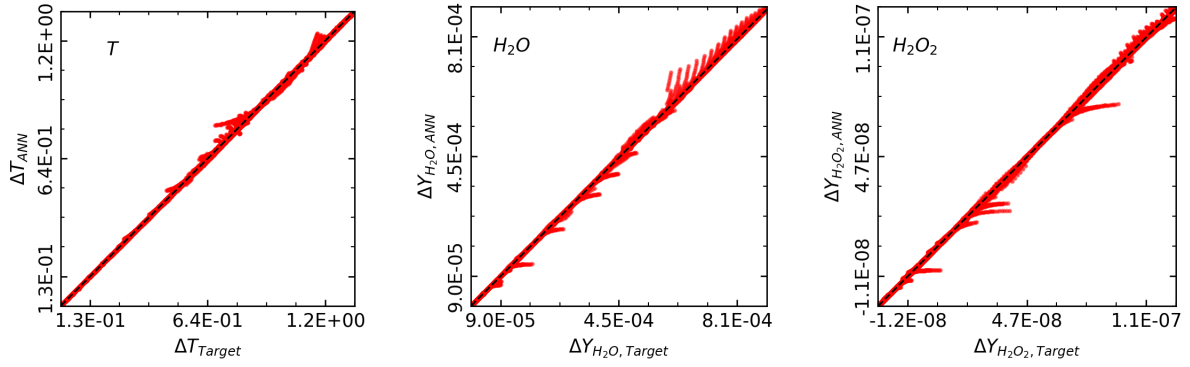


Figure 6: Comparison of the ANN a priori prediction with direct integration target values (black dashed line) using SOM data clustering.

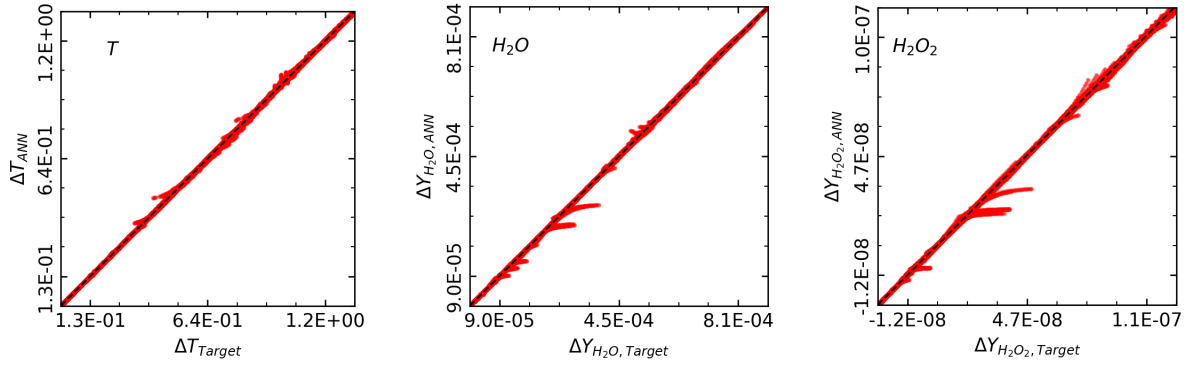


Figure 7: Comparison of the ANN a priori prediction with direct integration target values (black dashed line) using KMBDT data clustering.

than the previous investigation, as errors that are introduced into the model in the beginning can have an effect later on. The same range of starting conditions is employed for the chemical reactor as for the a priori benchmark. Finally, the ANN results are validated over the entire physical simulation time range against results from direct integration, which are obtained with Cantera. First, the ability to reproduce the ignition delay time (IDT) is evaluated, then exemplary trajectories of species mass fractions are compared and finally the NRMSE is used for a quantitative quality assessment over the whole $T - \phi$ validation range.

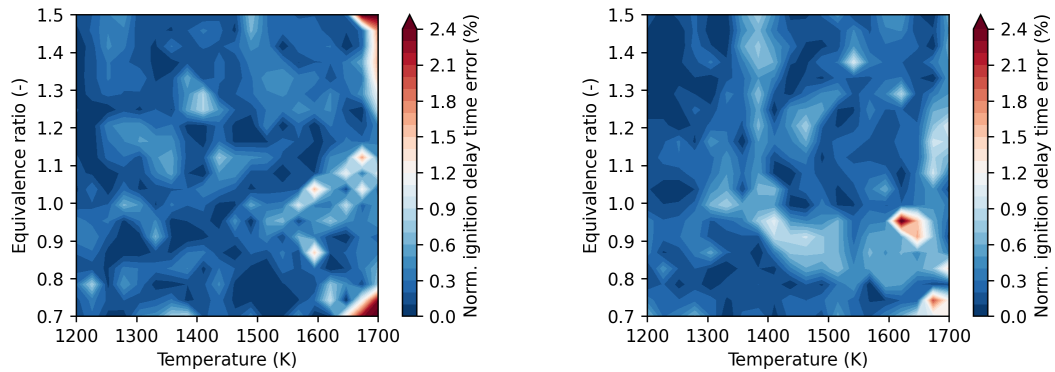


Figure 8: Normalized IDT error of SOM (left) and KMBDT (right).

Figure 8 presents the error in IDT prediction of both ANN compounds. The IDT τ is defined as

$$\tau = \arg \max_t \frac{\partial T}{\partial t} . \quad (23)$$

Both approaches predict the IDT remarkably well, with errors below 1 % over most of the validation $T - \phi$ range. The SOM-ANN features slightly higher errors in the marginal regions where high starting temperatures are prescribed in combination with high or low equivalence ratios. This is less pronounced for the KMBDT approach. However, the results show clearly that both methods are capable of predicting the IDT over a wide range of starting conditions.

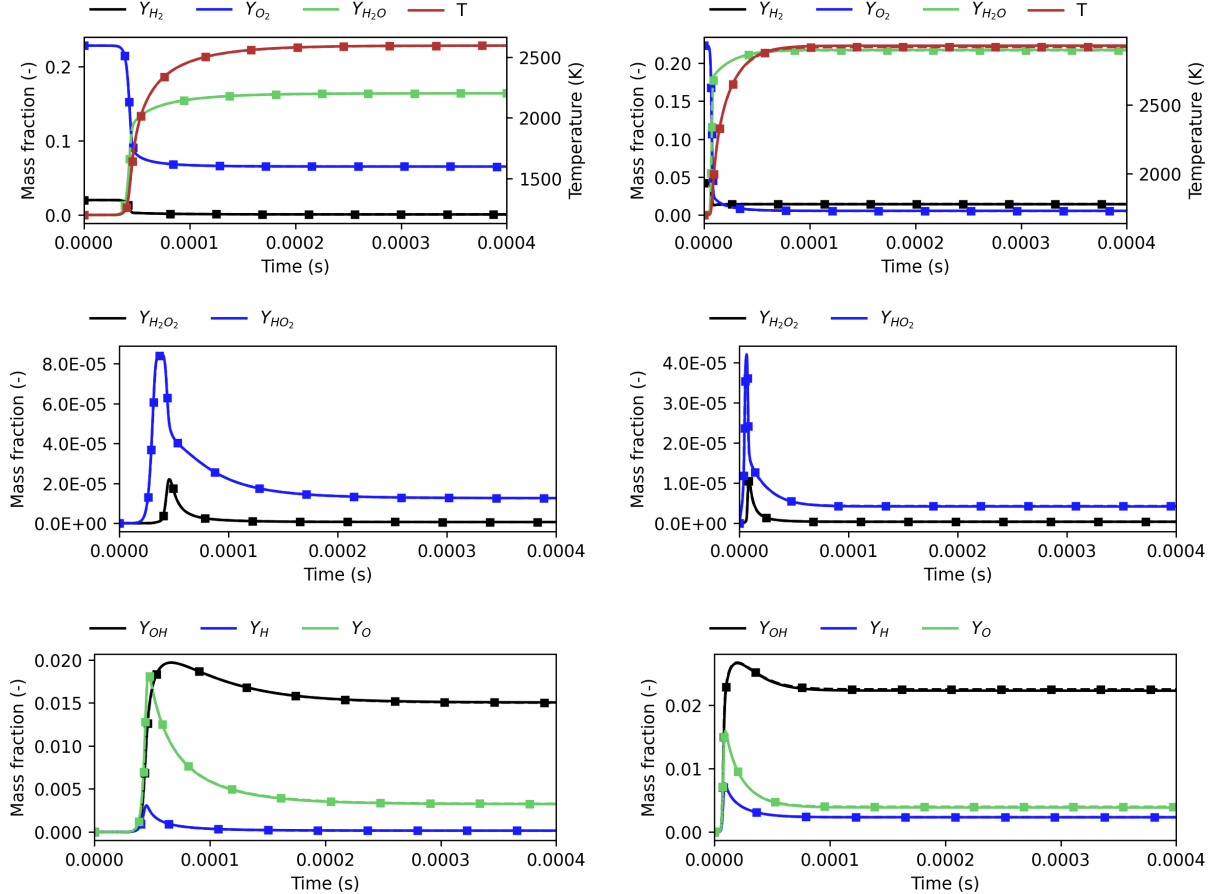


Figure 9: Temporal evolution of all quantities predicted by the ANN (solid line) and via direct integration (dashed line with symbols) at two different starting conditions. Due to the good agreement of the data, almost all the dashed lines are covered by the solid lines. Left: $T = 1200$ K and $\phi = 0.7$; right: $T = 1700$ K and $\phi = 1.5$.

Figure 9 shows the temporal evolutions of temperature and all species mass fractions predicted with the SOM approach and with Cantera via direct integration. The conditions are chosen to demonstrate the ability of the SOM-ANN compound to give accurate predictions at both lean and rich, as well as low and high temperature conditions. For brevity, the respective results of the KMBDT-ANN are not shown, since these are almost identical in both cases. At both conditions, no difference between the results predicted by the SOM-ANN and by direct integration can be observed. The ANN is able to predict maximum and final species concentrations and temperature with great accuracy. It is remarkable that all species of the mechanism are predicted well. Also the different dynamics i.e. a slow ascend or a fast rise in mass fraction, are predicted well. Despite of the already mentioned complications at the low temperature margin of the training data, the results clearly demonstrate the general ability of the chosen ANN compounds to calculate chemical reactors, not only for major species but for all species of a chemical mechanism.

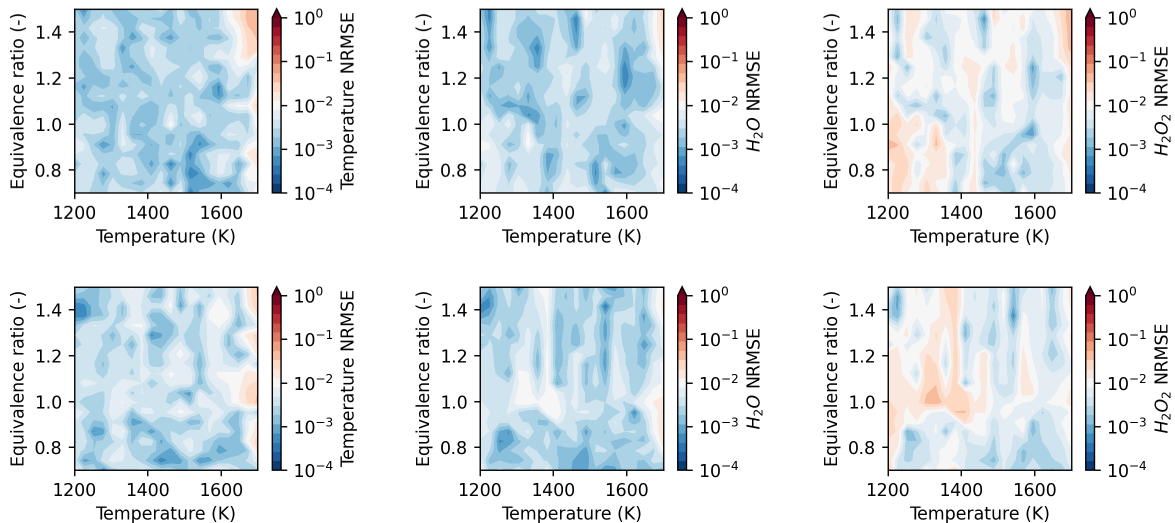


Figure 10: NRMSE of Temperature, H_2O - and H_2O_2 mass fraction prediction of the SOM-ANN (top) and the KMBDT-ANN (bottom).

A more comprehensive analysis and a quantitative a posteriori validation of the ANN-compounds can be done by the NRMSE. Figure 10 presents the NRMSE of temperature, H_2O and H_2O_2 for both ANN compounds over the entire validation range. All trends observed in the a priori validation and the IDT prediction can be observed here as well. The quality of the ANN prediction is very good for most of the validation range, but decreases towards the temperature maximum and minimum. Errors are very low for the temperature and the combustion product H_2O and higher for the intermediate species H_2O_2 . Interestingly, the error for the latter tends to increase in the low temperature range. Both the SOM-ANN and the KMBDT-ANN show the same trends and error magnitudes. With this comprehensive analysis it is finally possible to conclude that the investigated ANN compounds are valid estimators for a chemical reactor system. In particular the ability to predict not only the major species, but also the minor species like radicals is demonstrated.

D. Computational costs

The main goal of the introduction of ANNs instead of using direct integration is to save computational time. This assumption is verified in this section by benchmarking both the SOM-ANN and the KMBDT-ANN as well as Cantera. In total 400 chemical reactors have been calculated for the comparison. Table 3 shows the mean simulation time of a chemical reactor and the relative speedup. All computations have been performed on a single Intel® Core™ i7-8700 processor with 3.20 GHz. The SOM-ANN outperforms Cantera by a factor of 2.37. In contrast, the KMBDT-ANN is even faster with a performance increase by a factor of 9.00. Thus, the KMBDT-ANN is more than three times as fast as the SOM-ANN, although it consists of a larger number of MLPs. Since the exact same MLP architecture is used for both compounds, the difference is due to the clustering methods. To choose the best-matching MLP, the SOM-ANN needs to compute the distance between the input values and all 875 SOM nodes at every iteration. The KMBDT is more efficient because the maximum amount of distance calculations equals the depth of the tree, which corresponds to a maximum of 13 layers in this work. This large difference in the number of distance calculations results in the better performance of the KMBDT-ANN compound.

IV. Conclusions

In this work, a framework for an efficient neural network approach that should replace the computationally tedious direct integration in combustion problems was developed. The application of the methodology consists of two steps. First, a classification of the thermochemical state space into different clusters is

Table 3: Runtime comparison of ODE and ANN solvers.

| Approach | mean simulation time (s) | relative speedup |
|---------------|--------------------------|------------------|
| Cantera (ODE) | 0.45 | 1.00 |
| SOM ANN | 0.19 | 2.37 |
| KMBDT ANN | 0.05 | 9.00 |

employed. This is followed by the application of a MLP for each cluster. Two methods are tested to perform the clustering, namely SOM and KMBDT. SOM has been used in various previous works of other groups and has shown good results in combination with MLPs so far. KMBDT, on the other hand, uses a binary decision tree structure in order to separate the thermochemical state space. This has the advantage that the determination of the best-matching unit is, in most cases, more efficient than SOM and thus computing time can be saved. A particular property of the employed methodology is the consideration of all species, which is necessary for a correct prediction of IDTs.

The methodology was tested with a hydrogen oxidation mechanism that contains eight species. Training data is generated using an adiabatic isobaric perfectly stirred reactor model at different starting temperatures and starting equivalence ratios. The data is clustered using one of the two mentioned clustering methods and, subsequently, one MLP is trained for each cluster. About 1170 clusters are utilized for the KMBDT and 875 for the SOM clustering method. The a priori validation for both SOM-ANN and KMBDT-ANN shows very good results for the majority of the species. Only for the minor species, especially H_2O_2 , slightly larger deviations occur. In predicting the evolution of a complete chemical reactor, both methods generate thermochemical state space trajectories and corresponding IDTs that fit remarkably well to results obtained by direct integration. This is achieved over a temperature range from 1200 K to 1700 K and equivalence ratios from 0.7 to 1.5. However, greater deviations appear for lower temperatures, which are close to the limit of the training interval, especially at lower temperatures. It turns out that these errors are majorly attributed to an incorrect prediction of minor species. This highlights the importance of an appropriate data normalization technique for both the clustering and the prediction via MLPs.

In addition to the excellent results with various specified starting conditions, speedups of 2.4 and 9.0 in comparison to an established direct integration method are achieved using SOM-ANN and KMBDT-ANN, respectively. For mechanisms that include more species and reactions it is expected that the speedups are even higher. Furthermore, the performance could be increased even further by training the ANN compound for larger computational time steps. This greatly illustrates the time saving potential of the presented method, when it is combined with a CFD solver for detailed combustion simulations of technical applications. Future works need concentrate on improvements in the training process and the selection of the training data set, so that the integration of the method in a CFD solver can be achieved.

References

- ¹Perrault, R., Shoham, Y., Brynjolfsson, E., Clark, J., Etchemendy, J., Grosz, B., Lyons, T., Manyika, J., Mishra, S., and Nibbles, J. C., "The AI Index 2019 Annual Report," Tech. rep., AI Index Steering Committee, Human-Centered AI Institute, Stanford University, Stanford, 2019.
- ²Ihme, M., Schmitt, C., and Pitsch, H., "Optimal artificial neural networks and tabulation methods for chemistry representation in LES of a bluff-body swirl-stabilized flame," *Proceedings of the Combustion Institute*, Vol. 32, No. 1, 2009, pp. 1527–1535.
- ³Kempf, A., Flemming, F., and Janicka, J., "Investigation of lengthscales, scalar dissipation, and flame orientation in a piloted diffusion flame by LES," *Proceedings of the Combustion Institute*, Vol. 30, No. 1, 2005, pp. 557–565.
- ⁴Dalakoti, D. K., Wehrfritz, A., Savard, B., Day, M. S., Bell, J. B., and Hawkes, E. R., "An a priori evaluation of a principal component and artificial neural network based combustion model in diesel engine conditions," *Proceedings of the Combustion Institute*, Vol. 38, No. 2, 2021, pp. 2701–2709.
- ⁵Lapeyre, C. J., Misdariis, A., Cazard, N., Veynante, D., and Poinot, T., "Training convolutional neural networks to estimate turbulent sub-grid scale reaction rates," *Combustion and Flame*, Vol. 203, 2019, pp. 255–264.
- ⁶Nikolaou, Z., Chrysostomou, C., Vervisch, L., and Cant, S., "Progress variable variance and filtered rate modelling using convolutional neural networks and flamelet methods," *Flow, Turbulence and Combustion*, Vol. 103, No. 2, 2019, pp. 485–501.
- ⁷Christo, F., Masri, A., and Nebot, E., "Artificial neural network implementation of chemistry with PDF simulation of H_2/CO_2 flames," *Combustion and Flame*, Vol. 106, No. 4, 1996, pp. 406–427.
- ⁸Blasco, J., Fueyo, N., Dopazo, C., and Ballester, J., "Modelling the temporal evolution of a reduced combustion chemical system with an artificial neural network," *Combustion and Flame*, Vol. 113, No. 1-2, 1998, pp. 38–52.

- ⁹Blasco, J. A., Fueyo, N., Dopazo, C., and Chen, J., "A self-organizing-map approach to chemistry representation in combustion applications," *Combustion Theory and Modelling*, Vol. 4, 2000, pp. 61–76.
- ¹⁰Sen, B. A. and Menon, S., "Turbulent premixed flame modeling using artificial neural networks based chemical kinetics," *Proceedings of the Combustion Institute*, Vol. 32, No. 1, 2009, pp. 1605–1611.
- ¹¹Sen, B. A. and Menon, S., "Linear eddy mixing based tabulation and artificial neural networks for large eddy simulations of turbulent flames," *Combustion and Flame*, Vol. 157, No. 1, 2010, pp. 62–74.
- ¹²Chatzopoulos, A. and Rigopoulos, S., "A chemistry tabulation approach via Rate-Controlled Constrained Equilibrium (RCCE) and Artificial Neural Networks (ANNs), with application to turbulent non-premixed CH₄/H₂/N₂ flames," *Proceedings of the Combustion Institute*, Vol. 34, No. 1, 2013, pp. 1465–1473.
- ¹³Franke, L. L., Chatzopoulos, A. K., and Rigopoulos, S., "Tabulation of combustion chemistry via Artificial Neural Networks (ANNs): Methodology and application to LES-PDF simulation of Sydney flame L," *Combustion and Flame*, Vol. 185, 2017, pp. 245–260.
- ¹⁴Seltz, A., Domingo, P., and Vervisch, L., "Solving the population balance equation for non-inertial particles dynamics using probability density function and neural networks: Application to a sooting flame," *Physics of Fluids*, Vol. 33, No. 1, 2021, pp. 013311.
- ¹⁵Wan, K., Barnaud, C., Vervisch, L., and Domingo, P., "Chemistry reduction using machine learning trained from non-premixed micro-mixing modeling: Application to DNS of a syngas turbulent oxy-flame with side-wall effects," *Combustion and Flame*, Vol. 220, 2020, pp. 119–129.
- ¹⁶Kohonen, T., "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, Vol. 43, No. 1, 1982, pp. 59–69.
- ¹⁷Nguyen, H.-T., Domingo, P., Vervisch, L., and Nguyen, P.-D., "Machine learning for integrating combustion chemistry in numerical simulations," *Energy and AI*, Vol. 5, 2021, pp. 100082.
- ¹⁸MacQueen, J. et al., "Some methods for classification and analysis of multivariate observations," *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, Vol. 1, Oakland, CA, USA, 1967, pp. 281–297.
- ¹⁹Zhang, T., Zhang, Y., E, W., and Ju, Y., "DLODE: A deep learning-based ODE solver for chemical kinetics," *AIAA Scitech 2021 Forum*, *AIAA 2021-1139*, 2021.
- ²⁰Wang, F., Wang, Q., Nie, F., Li, Z., Yu, W., and Ren, F., "A linear multivariate binary decision tree classifier based on K-means splitting," *Pattern Recognition*, Vol. 107, 2020, pp. 107521.
- ²¹Gerlinger, P., *Numerische Verbrennungssimulation: Effiziente numerische Simulation turbulenter Verbrennung*, Springer-Verlag, 2005.
- ²²J. Warnatz, J. Maas, R. D., *Combustion: Physical and Chemical Fundamentals, Modeling and Simulation*, Vol. 4, Springer, 2006.
- ²³Kohonen, T., *Self-Organizing Maps*, Springer, Berlin, 2001.
- ²⁴Asan, U. and Ercan, S., "An introduction to self-organizing maps," *Computational Intelligence Systems in Industrial Engineering*, Springer, 2012, pp. 295–315.
- ²⁵Haykin, S., *Neural Networks and Learning Machines*, Vol. 3, Pearson, 2009.
- ²⁶Hastie, T., Tibshirani, R., and Friedman, J., *The elements of statistical learning: data mining, inference, and prediction*, Springer Science & Business Media, 2009.
- ²⁷Coates, A. and Ng, A. Y., "Learning feature representations with k-means," *Neural networks: Tricks of the trade*, Springer, 2012, pp. 561–580.
- ²⁸Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, Vol. 12, 2011, pp. 2825–2830.
- ²⁹Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al., "Pytorch: An imperative style, high-performance deep learning library," *arXiv preprint arXiv:1912.01703*, 2019.
- ³⁰Saxena, P. and Williams, F. A., "Testing a small detailed chemical-kinetic mechanism for the combustion of hydrogen and carbon monoxide," *Combustion and Flame*, Vol. 145, No. 1-2, 2006, pp. 316–323.
- ³¹"Chemical-Kinetic Mechanisms for Combustion Applications," San Diego Mechanism web page, Mechanical and Aerospace Engineering (Combustion Research), University of California at San Diego (<http://combustion.ucsd.edu>).
- ³²Goodwin, D. G., Moffat, H. K., and Speth, R. L., "Cantera: An Object-oriented Software Toolkit for Chemical Kinetics, Thermodynamics, and Transport Processes," <http://www.cantera.org>, 2017, Version 2.3.0.
- ³³Vettigli, G., "MiniSom: minimalistic and NumPy-based implementation of the Self Organizing Map," <https://github.com/JustGlowing/minisom/>, 2018.
- ³⁴Davies, D. L. and Bouldin, D. W., "A Cluster Separation Measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-1, No. 2, 1979, pp. 224–227.